

- d. Mampu mengelola *model base* dengan fungsi manajemen yang *analog* dengan manajemen data base (seperti mekanisme untuk menyimpan, membuat katalog, menghubungkan dan mengakses model).

3. Subsistem *dialog (user system interface)*

Keunikan lainnya dari SPK adalah adanya fasilitas yang mampu mengintegrasikan sistem terpasang dengan pengguna secara interaktif. Fasilitas atau subsistem ini dikenal sebagai subsistem *dialog*. Melalui sistem dialog inilah sistem diartikulasikan dan diimplementasikan sehingga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dirancang. Fasilitas yang dimiliki oleh subsistem ini dapat dibagi atas tiga komponen, yaitu:

- a. Bahasa aksi (*Action Language*), yaitu suatu perangkat lunak yang dapat digunakan pengguna untuk berkomunikasi dengan sistem. Komunikasi ini dilakukan melalui berbagai pilihan media seperti: *keyboard*, *joystick*, atau *key function* lainnya.
- b. Bahasa tampilan (*Display* atau *Presentation Language*), yaitu suatu perangkat yang berfungsi sebagai sarana untuk menampilkan sesuatu. Peralatan yang digunakan untuk merealisasikan tampilan ini diantaranya adalah *printer*, grafik *monitor*, *plotter* dan lain-lain.
- c. Basis Pengetahuan (*Knowledge Base*), yaitu bagian yang mutlak diketahui oleh pengguna sehingga sistem yang dirancang dapat berfungsi secara efektif.

2.2.2 Logika Fuzzy

Menurut Agus Naba, logika *fuzzy* adalah sebuah metodologi berhitung dengan variabel kata-kata (*linguistic variable*) sebagai pengganti berhitung dengan bilangan. Kata-kata digunakan dalam *fuzzy logic* memang tidak sepresisi bilangan, namun kata-kata jauh lebih dekat dengan intuisi manusia. Mengenai logika *fuzzy* pada dasarnya tidak semua keputusan dijelaskan 0 atau 1, namun ada kondisi diantara keduanya, daerah diantara keduanya inilah yang disebut dengan *fuzzy* atau tersamar. Secara umum ada beberapa konsep sistem logika *fuzzy*, sebagai berikut dibawah ini:

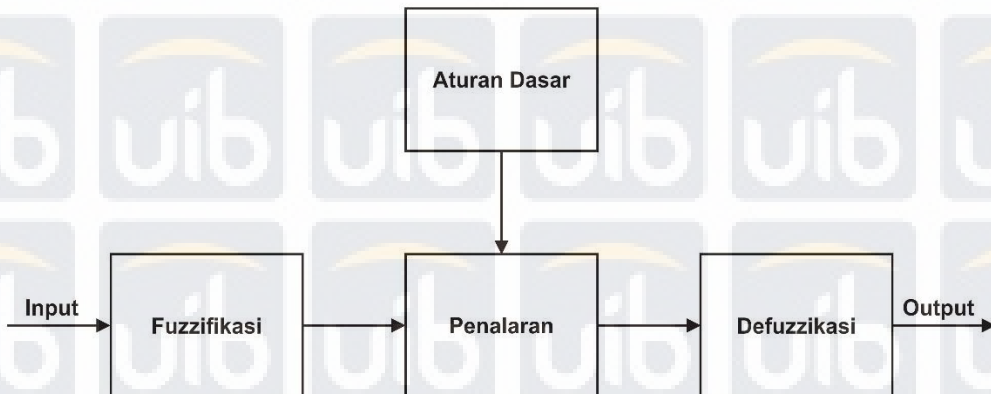
- a. Himpunan tegas yang merupakan nilai keanggotaan suatu *item* dalam suatu himpunan tertentu.
- b. Himpunan *fuzzy* yang merupakan suatu himpunan yang digunakan untuk mengatasi kekakuan dari himpunan tegas.
- c. Fungsi keanggotaan yang memiliki interval 0 dan 1.
- d. Variabel *linguistic* yang merupakan suatu variabel yang memiliki nilai berupa kata-kata yang dinyatakan dalam bahasa alamiah dan bukan angka.
- e. Operasi dasar himpunan *fuzzy* merupakan operasi untuk menggabungkan dan atau memodifikasi himpunan *fuzzy*.
- f. Aturan (*rule*) *if-then fuzzy* merupakan suatu pernyataan *if-then*, dimana beberapa kata-kata dalam pernyataan tersebut ditentukan oleh fungsi keanggotaan.

Dalam proses pemanfaatan logika *fuzzy*, ada beberapa hal yang harus diperhatikan salah satunya adalah cara mengolah *input* menjadi *output* melalui sistem inferensi *fuzzy*. Metode inferensi *fuzzy* atau cara merumuskan pemetaan dari masukan yang diberikan kepada sebuah keluaran. Proses ini melibatkan fungsi keanggotaan, operasi logika, serta aturan *IF-THEN*. Hasil dari proses ini akan menghasilkan sebuah sistem yang disebut dengan *FIS* (*Fuzzy Inference System*). Dalam logika *fuzzy* tersedia beberapa jenis *FIS* diantaranya adalah *Mamdani*, *Sugeno*, dan *Tsukamoto*.

Menurut Kusumadewi, S. dan Purnomo, H. (2010), ada beberapa alasan mengapa orang menggunakan logika *fuzzy* antara lain:

- a. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
- b. Logika *fuzzy* sangat fleksibel.
- c. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
- d. Logika *fuzzy* mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks.
- e. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
- f. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- g. Logika *fuzzy* didasarkan pada bahasa alami

2.2.3 Struktur Dasar Logika Fuzzy



Gambar 2.1 Hubungan Antar Sasaran Strategis

Sumber: [NUG-10:5]

Berdasarkan gambar 2.1, dalam sistem logika *fuzzy* terdapat beberapa tahapan operasional yang meliputi:

a. Fuzzifikasi

Fuzzifikasi adalah suatu proses perubahan nilai tegas yang ada ke dalam fungsi keanggotaan.

b. Penalaran (*Inference Machine*)

Mesin penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluaran sebagai bentuk pengambilan keputusan.

Salah satu model penalaran yang banyak dipakai adalah penalaran *max-min*. Dalam penalaran ini, proses pertama yang dilakukan adalah melakukan operasi min sinyal keluaran lapisan fuzzifikasi, yang diteruskan

dengan operasi *max* untuk mencari nilai keluaran yang selanjutnya akan difuzzifikasikan sebagai bentuk keluaran.

c. Aturan Dasar (*Rule Based*)

Aturan dasar (*rule based*) pada kontrol logika *fuzzy* merupakan suatu bentuk aturan relasi *if-then* seperti berikut ini:

if x is A then y is B dimana A dan B adalah *linguistic values* yang didefinisikan dalam rentang variabel X dan Y. Pernyataan “x is A” disebut *antecedent* atau premis. Pernyataan “y is B” disebut *consequent* atau kesimpulan.

d. Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu.

2.2.4 Metode Sugeno

Fuzzy metode sugeno merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk *IF – THEN*, dimana *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear [KUS-02:98].

Metode tersebut akan digunakan untuk menentukan jumlah pemesanan barang berdasarkan data persediaan barang dan jumlah permintaan. Data persediaan barang

dan jumlah permintaan adalah variabel-variabel yang akan direpresentasikan dengan fungsi keanggotaan *fuzzy*, selanjutnya metode *sugeno* untuk menentukan jumlah pemesanan barang diterapkan dalam Sistem Pendukung Keputusan (SPK). Kemudian SPK akan mengolah data-data tersebut dengan metode *sugeno* dan akan menampilkan keluaran (*output*) berupa jumlah barang yang akan dipesan.

Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Model *Sugeno* menggunakan fungsi keanggotaan *Singleton* yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada nilai *crisp* yang lain. Untuk *Orde 0* dengan rumus :

$$\text{IF } (x_1 \text{ is } a_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_n \text{ is } A_n) \text{ THEN } z = k \dots \dots \dots (1)$$

dengan A_i adalah himpunan *fuzzy* ke i sebagai *antaseden* (alasan), \circ adalah operator *fuzzy* (*AND* atau *OR*) dan k merupakan konstanta tegas sebagai konsekuensi (kesimpulan).

Sedangkan rumus *Orde 1* adalah:

$$\text{IF } (x_1 \text{ is } a_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_n \text{ is } A_n) \text{ THEN } z = p_1 * x_1 + \dots + p_n * x_n + q \quad (2)$$

dengan A_i adalah himpunan *fuzzy* ke i sebagai *antaseden*, \circ adalah operator *fuzzy* (*AND* atau *OR*), p_i adalah konstanta ke i dan q juga merupakan konstanta dalam konsekuensi. Metode *sugeno* menggunakan konstanta atau fungsi matematika dari variabel *input*:

$$\text{Jika } a \text{ adalah } \tilde{A}_i \text{ dan } b \text{ adalah } \tilde{B}_i, \text{ maka } c \text{ adalah } \tilde{C}_i = f(a,b) \dots \dots \dots (3)$$

Dengan a , b , dan c adalah variabel linguistik; \tilde{A}_i dan \tilde{B}_i himpunan *fuzzy* ke- i untuk a dan b , dan $f(a,b)$ adalah fungsi matematik.

Untuk mendapatkan *output* (hasil), maka terdapat 4 langkah / tahapan sebagai berikut:

1. Pembentukan himpunan *fuzzy*

Menentukan semua variabel yang terkait dalam proses yang akan ditentukan. Untuk masing-masing variabel *input*, tentukan suatu fungsi fuzzifikasi yang sesuai.

2. Aplikasi fungsi implikasi

Menyusun basis aturan, yaitu aturan-aturan berupa implikasi-implikasi *fuzzy* yang menyatakan relasi antara variabel *input* dengan variabel *output*.

3. Komposisi aturan

Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu Metode *Min (Minimum)*. Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai minimum aturan, kemudian menggunakan nilai tersebut untuk memodifikasi daerah *fuzzy* dan mengaplikasikannya ke *output* dengan menggunakan operator *AND*. Jika semua proporsi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proporsi. Secara umum dapat dituliskan:

$$\mu(x_i) = \min(\mu_{sf}(x_i), \mu_{kf}(x_i)) \dots \dots \dots (4)$$

dengan:

$\mu_{sf}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

4. Penegasan

Masukan dari proses penegasan adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan real yang tegas. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka dapat diambil suatu nilai tegas tertentu sebagai *output*. Apabila komposisi aturan menggunakan metode *sugeno* makna defuzzifikasi (Z^*) dilakukan dengan cara mencari nilai rata-rata terpusatnya.








$$Z^* = \frac{\sum_{i=1}^n d_i \mu_{\tilde{A}_i}(d_i)}{\sum_{i=1}^n \mu_{\tilde{A}_i}(d_i)} \dots\dots\dots(5)$$

Dengan d_i adalah nilai keluaran pada aturan ke- i dan $\mu_{\tilde{A}_i}(d_i)$ adalah derajat keanggotaan nilai keluaran pada aturan ke- i sedangkan n adalah banyaknya aturan yang digunakan.

2.2.5 Flowchart

Menurut Adelia (2011), *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara *sekuensial*. *Flowchart* digunakan untuk merepresentasikan maupun

mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

<i>Terminal</i>		<i>Process</i>	
<i>Flowline</i>		<i>Decision</i>	
<i>Input / Output</i>		<i>Internal Module Call</i>	
<i>External Module Call</i>			

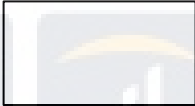



Tabel 2. 1 Simbol simbol *Flowchart*

2.2.6 *Data Flow Diagram (DFD)*

Data flow diagram adalah suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk dan simbol-simbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan. *Data flow diagram* ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem.

Dengan kata lain, *data flow diagram* adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. *Data flow diagram* ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan





sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program (Adelia, 2011).

Simbol	Nama	Keterangan
	Entitas	Objek aktif yang mengendalikan aliran data dengan memproduksi serta mengkonsumsi data.
	Proses	Objek yang melakukan transformasi terhadap data.
	Aliran Data	Aliran data menghubungkan keluaran dari suatu objek atau proses yang terjadi pada suatu masukan.
	<i>Data Store</i>	Objek pasif dalam <i>DFD</i> yang menyimpan data untuk penggunaan lebih lanjut.

Tabel 2. 2 Simbol simbol *DFD*

2.2.7 Entity Relationship Diagram (ERD)

Menurut Adelia (2011), *Entity Relationship Diagram* (ERD) yaitu model konseptual yang menjabarkan hubungan antar penyimpanan data dan hubungan data. Pada *Entity Relationship Diagram* (ERD) terdapat simbol-simbol dengan himpunan relasi yang masing-masing memiliki atribut untuk menjelaskan suatu relasi secara keseluruhan atau melakukan aktivitas permodelan data.

No	Simbol	Keterangan
1	Entitas 	Entitas adalah sebuah kesatuan objek lain, setiap entitas dibatasi oleh atribut.
2	Atribut 	Atribut merupakan sifat atau karakteristik dari suatu entitas yang menyediakan penjelasan secara rinci.
3	Relasi 	Menyatakan himpunan relasi.
4	Link 	Link sebagai penghubung antara himpunan relasi dari himpunan entitas dengan atributnya.

Tabel 2. 3 Simbol simbol ERD

2.2.8 Object Oriented Programming

Menurut Muhamad Danuri (2009), OOP (*Object Oriented Programming*) adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan tehnik berorientasi atau berbasis pada sebuah obyek dalam pembangunan program aplikasi, maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah Obyek yang didalamnya berisi beberapa perintah-perintah standar sederhana. Obyek ini dikumpulkan dalam Modul form atau Report atau modul lain dan disusun didalam sebuah project.

2.2.9 Database

Menurut Frederick Constantianus (2012), Basis Data (*database*) adalah kumpulan data yang diorganisasikan agar informasi yang terkandung didalamnya dapat dengan mudah diakses, dikelola serta diperbaharui. Basis data digunakan untuk menyimpan, memanipulasi dan mengambil data hampir semua tipe perusahaan termasuk bisnis, pendidikan, rumah sakit, pemerintahan dan perpustakaan.

2.2.10 Database Management System (DBMS)

Menurut Firdayanti, Meriza (2012), DBMS adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola dan mengendalikan pengaksesan basis data. Tujuan utama DBMS adalah menyediakan langkah yang nyaman dan efisien untuk penyimpanan dan pengambilan data dari basis data. DBMS berperan memberi abstraksi data tingkat tinggi ke pemakai.

2.2.11 Visual Basic .NET (VB)

Menurut Adelia dan Setiawan (2011), *Mircosoft Visual Basic* merupakan sebuah bahasa pemrograman yang bersifat *event driven* dan menawarkan *Integrated Development Environment* (IDE) visual untuk membuat aplikasi yang berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan *Data Access Objects* (DAO), *Remote Data Objects* (RDO) atau *ActiveX Data Object* (ADO), serta menawarkan pembuatan kontrol *ActiveX* dan objek *ActiveX*.

2.2.12 *Structured Query Language (SQL)*

Menurut (Adelia dan Setiawan, 2011), SQL adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. SQL terdiri dari dua bahasa, yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*. Implementasi DDL dan DML berbeda untuk tiap *Database Management System (DBMS)*, namun secara umum implementasi setiap bahasa ini memiliki bentuk standar yang ditepakan oleh ANSI (Yourness, 1991) yaitu antara lain:

1. *Data Definition Language (DDL)*

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya *table*, *view*, *user*, dan sebagainya. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data. Secara umum DDL yang digunakan adalah:

- a. *CREATE* untuk membuat objek baru.
- b. *USE* untuk menggunakan objek.
- c. *ALTER* untuk mengubah objek yang sudah ada.
- d. *DROP* untuk menghapus objek.

2. *Data Manipulation Language* (DML)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah-perintah yang umum dilakukan yaitu antara lain:

- a. *SELECT* untuk menampilkan data.
- b. *INSERT* untuk menambahkan data baru.
- c. *UPDATE* untuk mengubah data yang sudah ada.
- d. *DELETE* untuk menghapus data.

2.2.13 *Microsoft Visual Studio*

Menurut Rahadian (2011), Microsoft Visual Studio merupakan salah satu perangkat lunak buatan Microsoft yang dirancang khusus dalam pembuatan aplikasi-aplikasi profesional berbasis windows platforms. Dengan demikian, waktu untuk melakukan perubahan terhadap situs dengan jumlah halaman yang banyak dapat dikurangi berkat bantuan CSS.